CLAIMS

1. A method comprising:

sorting, using multiple depth buffers, depth data associated with multiple transparent pixels that overlie one another to identify an individual pixel that lies closest to an associated opaque pixel;

computing a transparency effect of the identified pixel relative to the associated opaque pixel; and

after said computing, identifying a next closest transparent pixel relative to the opaque pixel and computing, for the next closest pixel, a transparency effect relative to the transparency effect that was just computed.

2. The method of claim 1, wherein said multiple depth buffers comprise z buffers.

3. The method of claim 1, wherein said act of sorting comprises:

identifying one of the multiple buffers as a destination buffer that is both readable and writable;

identifying another of the multiple buffers as a source buffer that is only readable; and

flipping which of the multiple buffers is considered as the destination buffer and the source buffer during said acts of sorting, computing and identifying.

4. The method of claim 1 further comprising repeating said act of identifying for any additional overlying transparent pixels.

5. A computing system configured to implement the method of claim 1.

6. An apparatus comprising:

means for sorting, using multiple depth buffers, depth data associated with multiple transparent pixels that overlie one another to identify an individual pixel that lies closest to an associated opaque pixel;

means for computing a transparency effect of the identified pixel relative to the associated opaque pixel; and

means for identifying a next closest transparent pixel relative to the opaque pixel and computing, for the next closest pixel, a transparency effect relative to the transparency effect that was just computed.

7. The apparatus of claim 6, wherein said means for sorting and means for identifying comprises hardware comparison logic.

8. A method comprising:

(a) rendering at least one opaque pixel that lies along a ray;

(b) identifying a transparent pixel that lies along the ray, the identified transparent pixel being the closest transparent pixel to the opaque pixel;

(c) computing transparency effects of the identified transparent pixel relative to the opaque pixel;

(d) if additional transparent pixels lie along the ray, identifying a next closest transparent pixel relative to the opaque pixel and computing transparency effects of the next closest pixel relative to the computed transparency effects of a last computed transparent pixel; and

(e) repeating act (d) until transparency effects of all of the transparent pixels along the ray have been computed in a back-to-front manner.

9. The method of claim 8, wherein acts (a)-(e) are performed utilizing two physical depth buffers for sorting depth data associated with the transparent pixels.

10. The method of claim 8, wherein acts (a)-(e) are performed utilizing two physical depth buffers for sorting depth data associated with the transparent pixels, and wherein the two depth buffers are configured to be flipped.

11. The method of claim 8, wherein acts (a)-(e) are performed utilizing two physical z buffers for sorting depth data associated with the transparent pixels.

12. The method of claim 8, wherein acts (a)-(e) are performed utilizing two physical z buffers for sorting depth data associated with the transparent pixels, and wherein the two z buffers are configured to be flipped.

13. The method of claim 8, wherein acts (a)-(e) are performed utilizing two physical depth buffers for sorting depth data associated with the transparent pixels, and wherein performing acts (a)-(e) comprise:

designating one of the depth buffers as readable and writable;

designating the other of the depth buffers as readable only; and

flipping the designations of the depth buffers.

14. The method of claim 8, wherein acts (a)-(e) are performed utilizing two physical z buffers for sorting depth data associated with the transparent pixels, and wherein performing acts (a)-(e) comprise:

designating one of the z buffers as readable and writable;

designating the other of the z buffers as readable only; and

flipping the designations of the z buffers.

15. A computing system configured to implement the method of claim 8.

16. A system comprising:

means for rendering at least one opaque pixel that lies along a ray;

means for identifying a transparent pixel that lies along the ray, the identified transparent pixel being the closest transparent pixel to the opaque pixel;

means for computing transparency effects of the identified transparent pixel relative to the opaque pixel; and

means for identifying, in a back-to-front manner, additional transparent pixels and successively computing transparency effects for the additional transparent pixels.

17. The system of claim 16, wherein said means for rendering comprises a graphics subsystem.

18. The system of claim 16, wherein said means for identifying comprises a pair of physical depth buffers.

19. The system of claim 16, wherein said means for identifying comprises a pair of physical depth buffers that can be logically flipped.

20. A system comprising:

a transparent depth sorting component comprising:

at least two physical depth buffers;

a writeback counter to count writebacks that occur to at least one of the two physical depth buffers; and

comparison logic that is configured to effect:

sorting, using said at least two physical buffers, of depth data associated with multiple transparent pixels that overlie one another to identify an individual pixel that lies closest to an associated opaque pixel;

computing a transparency effect of the identified pixel relative to the associated opaque pixel;

after said computing, identifying a next closest transparent pixel relative to the opaque pixel; and

computing, for the next closest pixel, a transparency effect relative to the transparency effect that was computed for the said closest individual pixel and the associated opaque pixel.

21. The system of claim 20, wherein:

one of said at least two physical depth buffers is capable of being designated as readable and writable;

another of said at least two physical depth buffers is capable of being designated as readable only; and

designations of said at least two physical depth buffers are capable of being flipped.

22. The system of claim 20, wherein said at least two physical depth buffers comprise z buffers.

23. The system of claim 20, wherein said transparent depth sorting component is configured to terminate transparent depth sorting when the writeback counter indicates that no writebacks have occurred.

24. A graphics subsystem embodying the transparent depth sorting component of claim 20.

25. A computer system embodying the graphics subsystem of claim 24.

26. A method comprising:

mapping a first of two depth buffers as a destination buffer that is readable and writable, a second of the two depth buffers being designated as a source buffer that is only readable;

rendering one or more opaque objects having associated opaque pixels;

writing a depth value associated with an opaque pixel to the first buffer;

mapping the second of the depth buffers as the destination buffer, the first of the depth buffers being designated as the source buffer;

initializing the destination buffer to a predetermined value;

effecting a comparison of a new pixel depth value with values in the source and destination buffers and writing the new pixel depth value to the destination buffer if the new pixel depth value is (a) greater than the value currently in the destination buffer and (b) less than the value in the source buffer, effective to write a new pixel depth value that is associated with a pixel that is closest to a pixel whose depth value is contained in the source buffer;

rendering one or more transparent objects having associated transparent pixels;

determining if transparency effects for all transparent pixels along an associated ray have been accounted for and if so, terminating processing for pixels that lie along the ray and, if not:

mapping the first of the depth buffers as the destination buffer, the second of the buffers being designated as the source buffer;

effecting a comparison of the new pixel depth value with values in the source and destination buffers and writing to a frame buffer and the destination buffer if the new pixel depth value is equal to the value in the

source buffer and the new pixel depth value is less than the value in the destination buffer;

rendering one or more transparent objects; and

returning to said act of mapping the second of the depth buffers until transparency effects of all transparent pixels that lie along the ray have been accounted for.

27. The method of claim 26, wherein said predetermined value comprises a depth buffer's smallest value.

28. The method of claim 26, wherein said act of determining is performed by maintaining a depth buffer writeback counter that keeps track of depth buffer writebacks.

29. The method of claim 26, wherein the depth buffers comprise z buffers.

30. A computing system configured to implement the method of claim 26.

31. A system comprising:

a processor;

at least two depth buffers;

a frame buffer; and

a graphics subsystem operably connected with the processor and configured to, under the influence of the processor:

map a first of the depth buffers as a destination buffer that is readable and writable, a second of the depth buffers being designated as a source buffer that is only readable;

render one or more opaque objects having associated opaque pixels;

write a depth value associated with an opaque pixel to the first buffer;

map the second of the depth buffers as the destination buffer, the first of the depth buffers being designated as the source buffer;

initialize the destination buffer to a predetermined value;

effect a comparison of a new pixel depth value with values in the source and destination buffers and write the new pixel depth value to the destination buffer if the new pixel depth value is (a) greater than the value currently in the destination buffer and (b) less than the value in the source buffer, effective to write a new pixel depth value that is associated with a pixel that is closest to a pixel whose depth value is contained in the source buffer;

render one or more transparent objects having associated transparent pixels;

determine if transparency effects for all transparent pixels along an associated ray have been accounted for and if so, terminate processing for pixels that lie along the ray and, if not:

map the first of the depth buffers as the destination buffer, the second of the buffers being designated as the source buffer;

effect a comparison of the new pixel depth value with values in the source and destination buffers and write to the frame buffer

and the destination buffer if the new pixel depth value is equal to the value in the source buffer and the new pixel depth value is less than the value in the destination buffer;

render one or more transparent objects; and

return to said mapping the second of the depth buffers until transparency effects of all transparent pixels that lie along the ray have been accounted for.

32. The system of claim 31, wherein said predetermined value comprises a depth buffer's smallest value.

33. The system of claim 31 further comprising a depth buffer writeback counter that keeps track of depth buffer writebacks.

34. The system of claim 31, wherein the depth buffers comprise z buffers.